IN THE CLAIMS:

This listing of claims will replace all prior listings of claims in the application.

1. (Currently Amended): A method of processing frames for a delegated TCP connection, comprising:

receiving a user buffer descriptor specifying a location of a plurality of user buffers in a first portion of memory that is allocated to an application program, wherein each user buffer is configured to store processed frame data for the delegated TCP connection;

processing a first portion of the frames using an offload unit to produce first processed frame data;

indicating in a field of the user buffer descriptor how many of the user buffers are accepted for the delegated TCP connection before the processed frame data is stored in one of the user buffers;

uploading the first processed frame data to a user buffer in the first portion of memory that is allocated to the application program;

processing a second portion of the frames using the offload unit to produce second processed frame data;

uploading the second processed frame data to a legacy buffer in a second portion of the memory that is allocated to a software driver configured to communicate between the offload unit and a TCP stack; and

~~indicating in a field of the user buffer descriptor how many of the user buffers store the first processed frame data for the delegated TCP connection;~~

processing the second processed frame data using the TCP stack executed on a CPU to produce third processed frame data.

2. (Original): The method of claim 1, further comprising determining whether a special case exists during the processing of each of the frames.

3. (Cancelled)

4. (Original): The method of claim 1, wherein at least a portion of the second processed frame data is partially processed frame header data.

5. (Original): The method of claim 1, wherein at least a portion of the second processed frame data is payload data.

6. (Cancelled)

7. (Cancelled)

8. (Original): The method of claim 1, further comprising finishing processing of the second processed frame data by the TCP stack executed on the CPU.

9. (Currently Amended): A system for processing data for a delegated TCP connection, comprising:

a TCP stack configured to process received frames stored in at least one legacy buffer;

a software driver configured to interface between the TCP stack and an offload unit;

a memory configured to store user buffers in a first portion that is allocated to an application program and to store legacy buffers in a second portion that is allocated to the software driver, wherein each user buffer is configured to store processed frame data for the delegated TCP connection; and

the offload unit configured to:

receive user buffer descriptors specifying locations of the user buffers;

indicate in a field of the user buffer descriptor how many of the user buffers are accepted to store payload data for the delegated TCP connection before the processed frame data is stored in one of the user buffers;

process frames received on the delegated TCP connection to produce payload data and partially processed frames;

upload partially processed frames to at least one of the legacy buffers; and

upload the payload data to at least one of the user buffers.

10. (Cancelled)

11. (Currently Amended): The system of claim 9, wherein the offload unit is configured to notify the TCP stack when a threshold value specified for frames transmitted from the offload unit on the delegated connection is exceeded.

12. (Original): The system of claim 9, wherein the TCP stack is configured to process frames for which a special case exists.

13. (Cancelled)

14. (Cancelled)

15. (Previously Presented): The system of claim 9, wherein the offload unit is configured to receive additional frames while uploading payload data to one of the legacy buffers or uploading processed frames to one of the user buffers.

16. (Currently Amended): A method of processing frames for delegated and non-delegated TCP connections, comprising:

receiving user buffer descriptors that specify locations of a plurality of user buffers in a first portion of memory that is allocated to an application program, wherein each user buffer is configured to store processed frame data for the delegated TCP connections;

processing delegated TCP connections using an offload unit, the offload unit configured to process frames for which special cases do not exist to produce processed frame data;

indicating in a field of a user buffer descriptor how many of the user buffers are accepted to store processed frame data for each one of the delegated TCP connections before the processed frame data is stored in one of the user buffers;

uploading the processed frame data to a user buffer in a first portion of memory that is allocated to an application program;

processing non-delegated TCP connections using a TCP stack executing on a CPU;

uploading frame data for the non-delegated TCP connections to a legacy buffer in a second portion of the memory that is allocated to a software driver configured to communicate between the offload unit and the TCP stack; and

processing all frames for which special cases exist using the TCP stack executing on the CPU.

17. (Cancelled)

18. (Cancelled)

19. (Previously Presented): The method of claim 16, further comprising updating connection state information stored in the offload unit when data is received for a delegated TCP connection.

20. (Previously Presented): The method of claim 16, further comprising updating connection state information stored in the offload unit when data is transmitted for a delegated TCP connection.

21-25 (Cancelled)

26. (Previously Presented): The method of claim 1, further comprising determining that a sync request flag is not asserted, the sync request flag controlling flushing of existing user buffer descriptors for user buffers that do not store processed frame data and discarding of new user buffer descriptors for the delegated TCP connection that are received while the sync request flag is asserted.

27. (Currently Amended): The method of claim 1, further comprising:

determining that a threshold value specified for frames transmitted from the

offload unit on the delegated TCP connection is exceeded; and

setting a flag in a notification field indicating that the threshold value is exceeded.

28. (Previously Presented): The system of claim 9, further comprising storing the user buffer descriptors in a connection buffer table entry corresponding to the delegated TCP connection, wherein the connection buffer table entry includes a sync flag that controls flushing of existing user buffer descriptors for user buffers that do not store processed frame data and discarding of new user buffer descriptors for the delegated TCP connection that are received while the sync request flag is asserted.

29. (Previously Presented): The method of claim 16, further comprising determining that a sync request flag is not asserted for a delegated TCP connection, the sync request flag controlling flushing of existing user buffer descriptors for user buffers that do not store processed frame data and discarding of new user buffer descriptors for the delegated TCP connection that are received while the sync request flag is asserted.

30. (Currently Amended): The method of claim 16, further comprising:

determining that a threshold value specified for frames transmitted from the offload unit on a delegated TCP connection is exceeded; and

setting a flag in a notification field indicating that the threshold value is exceeded for the delegated TCP connection.

31. (New): The method of claim 1, further comprising the step of issuing a notification command after the first processed frame data is uploaded to the user buffer.

32. (New): The system of claim 9, wherein the offload unit is further configured to issue a notification command after the payload data is uploaded to at least one of the user buffers.

33. (New): The method of claim 16, further comprising the step of issuing a notification command after the processed frame data is uploaded to the user buffer.